# UE Statistische Mustererkennung
# WS 2021
# Angaben zur ersten Aufgabengruppe

# 1 Aufgabe UE-I.1

Implementieren Sie die Funktion y_te = kNN(X_tr, y_tr, X_te, k), welche – gegeben die Trainingsmenge (Merkmale, Label) = (X_tr, y_tr ) – mittels kNN die Klassen-Labels für die Test-Merkmalsvektoren in X_te berechnet. Sie können annehmen, dass es nur 2 Klassen gibt.

a) Unterteilen Sie die Daten in den Eingabedateien (Merkmale, Label) = (*perceptrondata.txt perceptrontarget2.txt)* wiederholt, mindestens jedoch 5 mal, in eine jeweils gleich große Test- und Trainingsmenge. Ermitteln Sie Test- und Trainingsfehler für verschiedene Werte von k (z.B [1, 3, 5, .., 17 ]), und stellen Sie diese in geeigneter Form dar.

b) Bestimmen Sie für jede der obigen Aufteilungen in Test- und Trainingsmenge den optimalen Wert für $k$ mittels 5- und 10-facher Kreuzvalidierung.

Hinweise:

- Die Label (*targets*) sind sind zeilenweise gespeichert, die Merkmalsvektoren sind 2-dimensional. Die Eingabedateien können z.B. in Python mit der numpy-Funktion *loadtxt* oder in MATLAB mit der Funktion *dlmread* geladen werden.
- Die Label sind 0/1-codiert.

# 2 Aufgabe UE-I.2

Bezeichne $X$ die geworfene Augenzahl eines fairen, 6-seitigen Würfels. Bezeichne weiters $A$ das Ereignis $X \geq 4$ und $B$ das Ereignis $gerade(X)$.

a) Berechnen Sie $P(A \cup B)$ mittels der Summmenregel.

b) Berechnen Sie die $2 \times 2$ Kontingenztafel bzg. der obigen Ereignisse. Diese lässt sich auch als Kontingenztafel zweier abgeleiteter boolscher Zufallsvariablen mit $Y = X \geq 4$ und $Z = gerade(X)$ auffassen. Sind $Y$ und $Z$ unabhängig?

## 3 Aufgabe UE-I.3

Die Wahrscheinlichkeit, dass bei $n$ unabhängigen Bernoulli-Versuchen mit Parameter $\theta$ $r$ günstige Ausfälle beobachtet werden, ist bekanntlich durch die Binomialverteilung

$$P(X = r) = B(r|n, \theta) = \left( \begin{array}{c} n \\ r \end{array} \right) \theta^r (1 - \theta)^{n-r}$$

gegeben. Bezeichne $X$ die Anzahl der günstigen, $Y$ die Anzahl der ungünstigen Ausfälle. Erstellen Sie für $n = 3, \theta = 0.7$ die Tabelle der Verbundwahrscheinlichkeiten $P(X = r, Y = l), 0 \leq r, l \leq 3$. Sind $X$ und $Y$ unabhängig?

## 4 Aufgabe UE-I.4

Erstellen Sie eine Funktion, welche für die Pareto-Verteilung mit DF

$$p(x) = \frac{\alpha \ x_{min}^{\alpha}}{x^{\alpha+1}}, \ x \geq x_{min}$$

den "laufenden" Erwartungswert

$$\int_{x_{min}}^{x} x' \ p(x') \ dx'.$$

berechnet.

Bezeichne $x_q$ das $q$-Quantil der Verteilung, so gibt

$$L(q) = \frac{\int_{x_{min}}^{x_q} x' \ p(x') \ dx'}{\int_{x_{min}}^{\infty} x' \ p(x') \ dx'}$$

den Anteil der $q$ kleinsten Merkmalsausprägungen am Erwartungswert an. $L$ als Funktion von $q$ bezeichnet man als *Lorenz-Kurve*. Plotten Sie diese für $x_{min=1}, \alpha = 1.16, q \in [0, 1]$: der Punkt $(0.8, .2)$ sollte auf der Kurve liegen. Dies illustriert das sogenannte *Pareto-Prinzip*, wonach 80% der Population für nur 20% der gesamten Merkmalssumme verantwortlich sind . Wenn $x$ das Einkommen bezeichnet, heißt dies, dass 80% der Population nur 20% des gesamten Einkommens erhalten.

Hinweise:

- Für die Berechnung des mit $q$ korrespondierenden Quantilwerts dürfen Sie eine Bibliotheksfunktion verwenden. Die Quantilfunktion wird im Englischen oft als *percent point function, ppf* bezeichnet, unter Python/scipy finden Sie die entsprechende Funktion z.B. unter `scipy.stats.pareto.ppf`

- Der (laufende) Erwartungswert lässt sich für die Pareto-Verteilung am einfachsten durch symbolische Integration des Ausdrucks $p(x)\ x$ ermitteln, Sie können aber auch numerisch integrieren.